

Solutions ​ to​ ​ Sample​ ​ JAC444​ ​ Midterm​ ​ Test​ ​ ​1 ​ -2016​

**A.**​​**Theory**​(10​ ​marks ​​=​ ​5​ ​marks​ ​+​ ​5​ ​marks)

1. When ​ can​ ​ one​ ​ implement​ ​ a​ ​deep​​ ​cloning​ ​in​ ​Java?

see​ ​deep​ ​cloning​ ​lecture​ ​3

2.​ ​What ​ are​ ​ the​ ​ differences​ ​ between​ ​***this***​ ​ and​ ​***this()***​​ ?​ see​ ​this()​ ​lecture2

**B.**​​**Code**​​**–**​​**Quiz**​40​ ​marks​ ​=​ ​10​ ​quizzes​ ​\*​ ​4​ ​marks​ ​(1​ ​mark​ ​for​ ​correct​ ​answer​ ​and​ ​3​ ​for​ ​explanation​**)**

1.​What​ ​will​ ​happen​ ​when​ ​you​ ​attempt​ ​to​ ​compile​ ​and​ ​run​ ​this​ ​code?

**abstract**​​**class**​​**A**​​**{**

​​**abstract**​​**public**​​**void** ​ **method1();**​

​​**public**​​**void**​​**method2()**​​**{**

​​**System.out.println("The**​​**second**​​**method");**

​​**}**

**}**

**public**​​**class**​​**B**​​**extends**​ **A**​​ **{**​

​​**public** ​ **static**​​ **void**​​ **main(String**​​ **argv[]){**​

​​**A**​​**e**​​**=**​​**new**​​**B();**

​​**e.**​​**method2();**

​​**}**

​ **public**​​ **void**​​ **method1(){**​

​​**System.out.println("The**​​**first**​​**method");**

​ **}**​

​ **public**​ ​​**void** ​ **method2(){**​

​​**method1**​​**();**

​} ​**}**

1) ​ The​ ​ code​ ​ will​ ​ compile​ ​ ​and ​ run,​​ printing​ ​ out​ ​ the​ ​​words ​ ​"The ​ first​​ method"​

2)​ ​The​ ​compiler​ ​will​ ​complain​ ​that​ ​the​ ​A​ ​class​ ​is​ ​an​ ​abstract​ class.​

3)​ The​ ​ code​ ​ will​ ​ compile​ ​ and​​ run,​ ​ ​printing ​ out​ ​ the​ ​ words​ ​ "The​ ​ second​ ​ method​ ​ "​

4)​ ​The​ ​compiler​ ​will​ ​complain​ ​about​ ​the​ ​statement​ A​ ​e​ ​=​ ​new​ ​B();

**ANSWER:**​​**A**​​**Polymorphism**

***Explanation***​:Given​ ​the​ ​following​ ​code,​ ​what​ ​will​ ​happen​ ​when​ ​you​ ​try​ ​to​ ​compile​ ​and​ ​run​ ​it?

**public** ​ **class**​​ **Q2**​​ **{**​ **public**​​**static**​​**void**​​**main(String[]**​​**args)**​​**{ boolean** ​ **b1**​​ **=**​​ **false;**​​ **int**​​ ​**val** ​ **=**​​ **1**​ **; if**​​**((b1**​​**==**​​**true)**​​**&&**​​**((val**​​**+=**​​**1)** ​ **==**​​ ​**2))** ​​**System.out.println("Good:**​​**"**​​**+**​​**val);**

​​**else**

​ **System.out.println("Bad:**​ ​ **"**​​ **+**​​ **val);**​

​​**}**

**}**

A.​ ​Compilation​ ​error, ​ attempting​ ​ ​to​ ​perform​ ​binary​ ​comparison​ ​on​ logical​​ ​data ​​type.

B.​ ​Compilation​ ​and​ ​output​ ​of​ ​"Good: ​ 1​ ".

C.​ ​Compilation​ ​and​ ​output​ ​of ​ "Bad:​ ​ 2​ ".

D.​ ​Compilation​ ​and​ output​​ ​of​ ​"Bad:​ ​1".

**ANSWER:\_D**​ **short**​​**circuit**​​**logical**​​**operator**​​**&&**​​**in**​​**java**

3​ ​Given​ ​the​ ​following​ ​code,​ ​what​ ​will​ ​be​ ​the​ ​output?

**class**​​**Int**​​**{**

​​**public**​​**int**​​**i**​​**=**​​**1;**

**}**

**public** ​ **class**​​ **Q3**​​ **{**​

​​**public**​​**static**​​**void**​​**main(String**​​**argv[]) {**

**Q3** ​ **t**​​ **=**​​ **new**​​ **Q3();**​

​ **t.first();**​

​​**}**

​​**public** ​ **void**​​ **first()**​​​**{**

​​**int**​​**i**​​**=**​​**2;**

​ **Int**​​ **v**​​ **=**​​ **new**​​ **Int();**​

​​**v.i**​​**=**​​**3;**

​​**second(v,** ​ **i);**​

​​**System.out.println(v.i);**

​​**}**

​​**public** ​​**void** ​ **second(Int**​​**v,**​​​**int**​​**i)**​​**{**

​ **i**​​ **=**​​ **0**​ **;**

​ **v.i**​​ **=**​​ **4**​ **;**

​​**Int** ​ **val**​​ **=**​​​**new**​​**Int();**

​​**v**​​**=**​​**val;**

​​**System.out.println(v.i**​​**+**​​**"**​​**"** ​​**+**​​**i);**

​​**}**

**}**

1. 1 ​ 1​

4

1. 1 0

1

1. 4 0

4

1. 1 0

4

**ANSWER:\_D**​ **Polymorphism**​​**and**​​**field**​​**access**​​**in**​​**subclass**​​**(hidden)**

Given​ ​the​ ​following​ ​code,​ ​what​ ​will​ ​be​ ​the​ ​output?

**public**​​**class**​​**Q4**​​**{ public** ​ **static**​​ **void**​​ **main(String**​​ **args[])**​​​**{**

**System.out.println('e'**​​**-**​​**'b'**​​**+**​​**"A"**​​**+** ​​**4);**

**}**

**}**

A.​ ​3A4

B.​ ​'e'​ ​-​ ​'b'​ ​+​ ​"A"​ ​+​ ​4

C.​ Compilation​ ​ error​

D.​ ​None​ ​of​ ​these

# ANSWER:\_\_A​ ​String​ ​concatenation

2. What​ ​will​ ​happen​ ​when​ ​you​ ​try​ ​to ​compile​​ ​and​ ​run​ ​the​ ​following​ ​code?

**public**​​**class**​​**Test**​​**{**

​​**public**​​**void**​​**method()** ​ **{**​

​​**for(int**​​**i**​​**=**​​**0;**​​**i**​​**<**​​**3;**​​**i++)**​​**{**

​ **System.out.print(i);**​

​​**}**

​​**System.out.print(i);**

​​**}**

**}**

1. 0122
2. 0123
3. Compilation​ ​error
4. None​ ​of​ ​these

**ANSWER:\_\_C**

Test.java:6:​ ​error:​ ​cannot​ ​find​ ​symbol

​ ​System.out.print(i);

3. What​ ​is​ ​displayed​ ​when​ ​the​ ​following ​code​​ ​is​ ​compiled​ ​and ​​executed?

**String**​​**s1**​​**=**​​**new** ​ **String("Test");**​ **String**​​**s2**​​**=**​​**new**​​**String("Test"); if**​​**(s1==s2)**

​​**System.out.println("Same"); if**​​**(s1.equals(s2))**

​ **System.out.println("Equals");**​

1. Same

Equals

1. Equals
2. Same
3. The​ ​code​ ​compiles,​ ​but​ ​nothing ​is​ ​​displayed​ ​upon​ ​execution.
4. The​ ​code​ ​fails​ ​to​ ​compile

**ANSWER:\_\_B**​​**result**​​**of**​​**s1.equals(s2);**

4. What​ ​is​ ​displayed​ ​when​ ​the​ ​following ​code​​ ​is​ ​compiled​ ​and ​​executed?

**class**​​**Parent**​ **{**​

​​**private**​​**void**​​**method1()**​​**{**

​​**System.out.println("Parent's**​​**method1()");**

​ **}**​

​​**public** ​ **void**​​ **method2()**​​ **{**​

​ **System.out.println("Parent's**​ ​ **method2()");**​

​​**method1();**

​​**}**

**}**

**class**​​**Child**​​**extends** ​ **Parent**​​ **{**​

​​**public**​​**void**​ **method1()**​​ **{**​

​​**System.out.println("Child's**​​**method1()");**

​ **}**​

​​**public** ​ **static**​​ **void**​​ **main(String**​​ **args[]){**​

​ **Parent**​​ **p**​​ **=**​​ **new**​​ **Child();**​

​ **p.method2();**​

​​**}**

**}**

A. ​ Compile​ ​ time​ ​ error​

B.​ ​Run​ ​time​ ​error

C.​ ​prints​ ​: Parent's ​ method2()​

Parent's ​ method1()​

D.​ ​prints​ ​: Parent's ​​method2()

Child's​ method1()​

**ANSWER:\_\_C**​​**polymorphism**

# C.​ ​What​ ​does ​ the​ ​ following​ ​ code​ ​ print?​ ​ Please​ ​ ​explain ​ your​ ​ answer.​ ​ 20​ ​marks​ ​=​ ​5​ ​marks​ ​+​ ​15​ ​marks

**public**​​**class**​​**What**​​**{**

​​**public**​​**static**​​**void**​​**main(String[]**​​**args)** ​ **{**​

**System.out.println(1**​ **+**​​ **2**​​​**+**​​**"1**​​**+**​​**2"**​​**+**​​**1**​​**+**​​**2);**

​ **}**​

**}**

31​ ​+​ ​212​ ​string​ ​concatenation

**class**​​**Dog**​​**{**

​​**public** ​ **static**​​ **void**​​ **bark()**​​ **{**​

​​**System.out.print("woof**​​**");**

​ **}**​

**}**

**class**​​**Basenji**​​**extends**​​**Dog**​​**{** ​​**public** ​ **static**​​ **void**​​ **bark()**​​ **{**​

​​**System.out.println("how**​​**");**

​​**}**

**}**

**public** ​ **class**​​ **Bark**​​ **{**​

​​**public**​​**static** ​ **void**​​ **main(String**​​ **args[])**​​ **{**​

​​**Dog**​​**woofer** ​ **=**​​​**new**​​**Dog();**

​​**Dog**​ **nipper**​​ **=**​​ **new**​​ **Basenji();**​

​​**woofer.bark();**

​ **nipper.bark();**​

​​**}**

**}**

woof​ ​woof​ ​-​ ​polymorphism

**D.**​​**Code**​​**–**​​**Development**​(Question​ ​1​ ​=​ ​10​ ​marks​ ​Question​ ​2​ ​=​ ​20marks)

1.​ ​Consider​ ​the​ ​following​ ​Java​ ​expression

**x**​​**->**​​**{**​​**return**​​**(x**​​**>**​​**0**​​**&&**​​**x**​​**<**​​**10);**​​**}**

If​ ​this​ ​is​ ​a​ ​valid​ ​expression​ ​in​ ​Java,​ ​explain​ ​what​ ​it​ ​does​ ​and​ ​how​ ​would​ ​you​ ​use​ ​it.

wes​ ​-​ ​it​ ​is​ a​ ​ ​lambda ​ expression​ ​ -​ ​ it​ ​ takes​ ​ ​an​ integer​ ​ and​ ​ returns​ ​ boolean​ true​ ​if​ ​x​ ​is​ ​between​ ​1​ ​and​ ​10

2. Write​ ​a​ ​Java​ ​program ​ that​ ​ takes​ ​ two​ ​​arguments ​ on​ ​ a​ ​ command​ ​ line.​ ​ The​ ​ first​ ​ argument​ ​ is​ ​ the​ ​ name​ ​​of the​ ​text​ ​file,​ ​and​ ​the​ ​second​ ​is​ ​a​ ​string.​ ​Your​ ​program​ ​must​ ​read​ ​the​ ​file​ ​and​ ​print​ ​all​ ​the​ ​lines​ ​from​ ​the file​ ​where​ ​the​ ​given​ ​string​ ​is​ ​found.

Write​ ​another​ ​Java​ ​program ​ that​ ​ takes​ ​many​​ ​arguments ​ on​ ​ a​ ​ command​ ​ line.​ ​ All​ ​ arguments​ ​ are​ ​ ​the ​ text​ file​ ​names,​ ​except​ ​the ​​last,​ ​which ​ ​is ​ a​ ​ ​string.​ ​Find​ how​​ many​​ ​lines​ ​in​ ​a​ ​file ​​contain ​​the​ ​given​ string.​ Every​ ​file​ ​must​ ​be​ ​read​ ​in​ ​a​ ​different​ ​thread.

*Use*​​*in*​​*the*​​*second*​​*program*​​*as*​​*much*​​*code*​​*as*​​*you*​​*can*​​*from*​​*first*​​*program.*​​*Please*​​*properly*​​*document*​​*your*​​*code.*

see​ ​course​ ​sample​ ​Java​ ​IO